Improving Review Representations with User Attention and Product Attention for Sentiment Classification

Zhen Wu, Xin-Yu Dai,[*] **Cunyan Yin, Shujian Huang, Jiajun Chen**

National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, 210023, China
Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing, 210023, China
wuz@nlp.nju.edu.cn, {daixinyu,yincy,huangsj,chenjj}@nju.edu.cn

Abstract

Neural network methods have achieved great success in reviews sentiment classification. Recently, some works achieved improvement by incorporating user and product information to generate a review representation. However, in reviews, we observe that some words or sentences show strong user's preference, and some others tend to indicate product's characteristic. The two kinds of information play different roles in determining the sentiment label of a review. Therefore, it is not reasonable to encode user and product information together into one representation. In this paper, we propose a novel framework to encode user and product information. Firstly, we apply two individual hierarchical neural networks to generate two representations, with user attention or with product attention. Then, we design a combined strategy to make full use of the two representations for training and final prediction. The experimental results show that our model obviously outperforms other state-of-the-art methods on IMDB and Yelp datasets. Through the visualization of attention over words related to user or product, we validate our observation mentioned above.

Introduction

Sentiment analysis aims to determine people's attitudes towards some topic or the overall polarity to a document, interaction, or event. In recent years, sentiment analysis draws increasing attention of researchers and industries because of the rapid growth of online review sites such as Amazon, Yelp and IMDB. In this work, we focus on the task of document-level review sentiment classification, which is a fundamental task in the field of sentiment analysis and opinion mining (Pang and Lee 2008). The task aims to infer the overall sentiment intensity (e.g. 1-5 stars on the review site Yelp) of review documents written by users for products.

Dominating studies follow (Pang and Lee 2005; Pang, Lee, and Vaithyanathan 2002) and take sentiment classification as a special case of text classification problem. They usually regard user-marked sentiment polarities or ratings as labels and use machine learning algorithms to build sentiment classifiers with text features. Following the idea, many works devote to designing effective features from text (Pang,

---

[*]Corresponding author.

Lee, and Vaithyanathan 2002; Qu, Ifrim, and Weikum 2010) or additional sentiment lexicons (Ding, Liu, and Yu 2008; Taboada et al. 2011; Kiritchenko, Zhu, and Mohammad 2014). Motivated by the great success of deep learning in computer vision (Krizhevsky, Sutskever, and Hinton 2012), speech recognition (Dahl et al. 2012) and natural language processing (Bengio et al. 2003), more recent methods use neural networks to learn low-dimensional and continuous text representations without any feature engineering (Glorot, Bordes, and Bengio 2011; Socher et al. 2011; 2012; 2013; Kim 2014). These models achieve very competitive performances in sentiment classification.

Despite neural network based approaches have been quite effective for sentiment classification (Johnson and Zhang 2015; Tang, Qin, and Liu 2015a), they typically only focus on the text content while ignoring the crucial influences of users and products. It is a common sense that the user's preference and product's characteristic make a significant effect on the ratings. For different users, same word might express different emotional intensity. For example, a lenient user may use "good" to evaluate an ordinary product while a critical user might use "good" to express an excellent attitude. Similarly, product's characteristic also have an effect on review ratings. Reviews of high-quality products tend to receive higher ratings compared to those of low-quality products.

In order to incorporate user and product information into sentiment classification, Tang, Qin, and Liu (2015b) introduce a word-level preference matrix and a representation vector for each user and product into CNN sentiment classifier. The model achieves some improvements, but suffers from high model complexity and only considers word-level user and product information rather than semantic-level. Chen et al. (2016a) consider user and product information together and incorporate them into one review representation via attention mechanism (Bahdanau, Cho, and Bengio 2015). However, in reviews, we observe that some words or sentences show strong user's preference, and some others tend to indicate product's characteristic. For example, for the review "*the bar area is definitely good 'people watching' and i love the modern contemporary décor.*", the word "*good*", "*modern*" and "*contemporary*" describe the characteristic of product, and the "*love*" shows strong user's sentiment. Opinions are more related to products and emotions

are more centered on the user. They are called rational evaluation and emotional evaluation respectively by Liu (2012). Obviously, the two kinds of information have different effects on inferring sentiment label of the review. Intuitively, a review has different latent semantic representations with different views from users or products. Therefore, it is unreasonable to encode user and product information together into one representation.

In this paper, we address the above issues by encoding user and product information, respectively, into two hierarchical networks to generate two individual text representations with user attention or product attention. Then, we design a combined strategy to make most use of the two representations for training and final prediction, which proves effective. The experimental results show that our model obviously outperforms other state-of-the-art methods on IMDB and Yelp datasets. We open the source code in GitHub.[1]

The main contributions of our work are as follows:

- We propose a novel framework to encode review from two views for sentiment classification. With user attention and product attention, respectively, two representations are generated, which are concatenated for further classification.

- For better learning the neural network, we introduce a combined strategy to improve review representations. With a weighted loss function, better representations from two views can be achieved and further help sentiment classification.

- The experimental results demonstrate that our model achieves obvious and consistent improvements compared to all state-of-the-art methods. Some visualization cases also validate the effectiveness and interpretability of our method.

Background

Long Short-Term Memory

Long Short-Term Memory(LSTM) (Hochreiter and Schmidhuber 1997) is widely used for text modeling because of its excellent performance on sequence modeling, especially for long documents. In order to address the problem of long-term dependencies, the LSTM architecture introduces a memory cell that is able to preserve cell state over long periods of time.

There are three gates to protect and control the state flow in LSTM unit. At each time step t, given an input vector $\mathbf{x}_t$, the current cell state $\mathbf{c}_t$ and hidden state $\mathbf{h}_t$ can be updated with previous cell state $\mathbf{c}_{t-1}$ and hidden state $\mathbf{h}_{t-1}$ as follows:

$$\begin{bmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \end{bmatrix} (\mathbf{W}\left[\mathbf{h}_{t-1}; \mathbf{x}_t\right] + \mathbf{b}), \qquad (1)$$

$$\hat{\mathbf{c}}_t = \tanh\left(\mathbf{W}_c\left[\mathbf{h}_{t-1}; \mathbf{x}_t\right] + \mathbf{b}_c\right), \qquad (2)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \hat{\mathbf{c}}_{t-1} + \mathbf{i}_t \odot \hat{\mathbf{c}}_t, \qquad (3)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t), \qquad (4)$$

[1]https://github.com/wuzhen247/HUAPA

where $\mathbf{i}_t$, $\mathbf{f}_t$ and $\mathbf{o}_t$ are gate activations and in $[0, 1]$, σ is the logistic sigmoid function and $\odot$ stands for element-wise multiplication. Intuitively, the forget gate $\mathbf{f}_t$ controls the extent to which the previous memory cell is forgotten, the input gate $\mathbf{i}_t$ controls how much each unit is updated, and the output gate $\mathbf{o}_t$ controls the exposure of the internal memory state. The hidden state $\mathbf{h}_t$ denotes output information of LSTM unit's internal memory cell.

In order to increase the amount of input information available to the network, a more common approach is to adopt bidirectional LSTM to model text semantics both from forward and backward. For sequence vectors $[\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T]$, the forward LSTM reads sequence from $\mathbf{x}_1$ to $\mathbf{x}_T$ and the backward LSTM reads sequence from $\mathbf{x}_T$ to $\mathbf{x}_1$. Then we concatenate the forward hidden state $\overrightarrow{\mathbf{h}_t}$ and backward hidden state $\overleftarrow{\mathbf{h}_t}$, i.e., $\mathbf{h}_t = \left[\overrightarrow{\mathbf{h}_t}; \overleftarrow{\mathbf{h}_t}\right]$, where the $[\cdot; \cdot]$ denotes concatenation operation. The $\mathbf{h}_t$ summarizes the information of the whole sequence centered around $\mathbf{x}_t$.

Attention Mechanism

Inspired by human visual attention, the attention mechanism is proposed by Bahdanau, Cho, and Bengio (2015) in machine translation, which is introduced into the Encoder-Decoder framework to select the reference words in source language for words in target language. It is also used in image caption generation (Xu et al. 2015), parsing (Vinyals et al. 2015), natural language question answering (Sukhbaatar et al. 2015). Yang et al. (2016) and Chen et al. (2016a) explore hierarchical attention mechanism to select informative words or sentences for the semantics of document.

Document-level Sentiment Classification

Document-level sentiment classification usually aims to predict the corresponding sentiment label of review text. In general, a review is written by a user $u \in U$ for a product $p \in P$. We denote the review as a document d with n sentences $\{s_1, s_2, \cdots, s_n\}$ and l_i is the length of i-th sentence. The i-th sentence s_i consists of l_i words $\{w_{i1}, w_{i2}, \cdots, w_{il_i}\}$. For modelling document-level text semantics, we can employ bidirectional LSTM with hierarchical structure to obtain document representation. In word level, each word w_{ij} is mapped to its embedding $\mathbf{w}_{ij} \in \mathbb{R}^d$. BiLSTM network receives $[\mathbf{w}_{i1}, \mathbf{w}_{i2}, \cdots, \mathbf{w}_{il_i}]$ and generates hidden states $[\mathbf{h}_{i1}, \mathbf{h}_{i2}, \cdots, \mathbf{h}_{il_i}]$. Then we can fetch the last hidden state $\mathbf{h}_{il_i}$ or feed $[\mathbf{h}_{i1}, \mathbf{h}_{i2}, \cdots, \mathbf{h}_{il_i}]$ to an average pooling layer or use attention mechanism to obtain the sentence representation $\mathbf{s}_i$. In sentence level, we feed the generated sentence representations $[\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_n]$ into BiLSTM and then obtain the document representation $\mathbf{d}$ in a similar way. Finally, $\mathbf{d}$ is taken as feature of softmax classifier to predict sentiment label of the review.

Methods

It is obvious that not all words/sentences contribute equally to the review text semantics for different users and different products. In addition, in reviews, we observe that some words or sentences show strong user's preference, and some
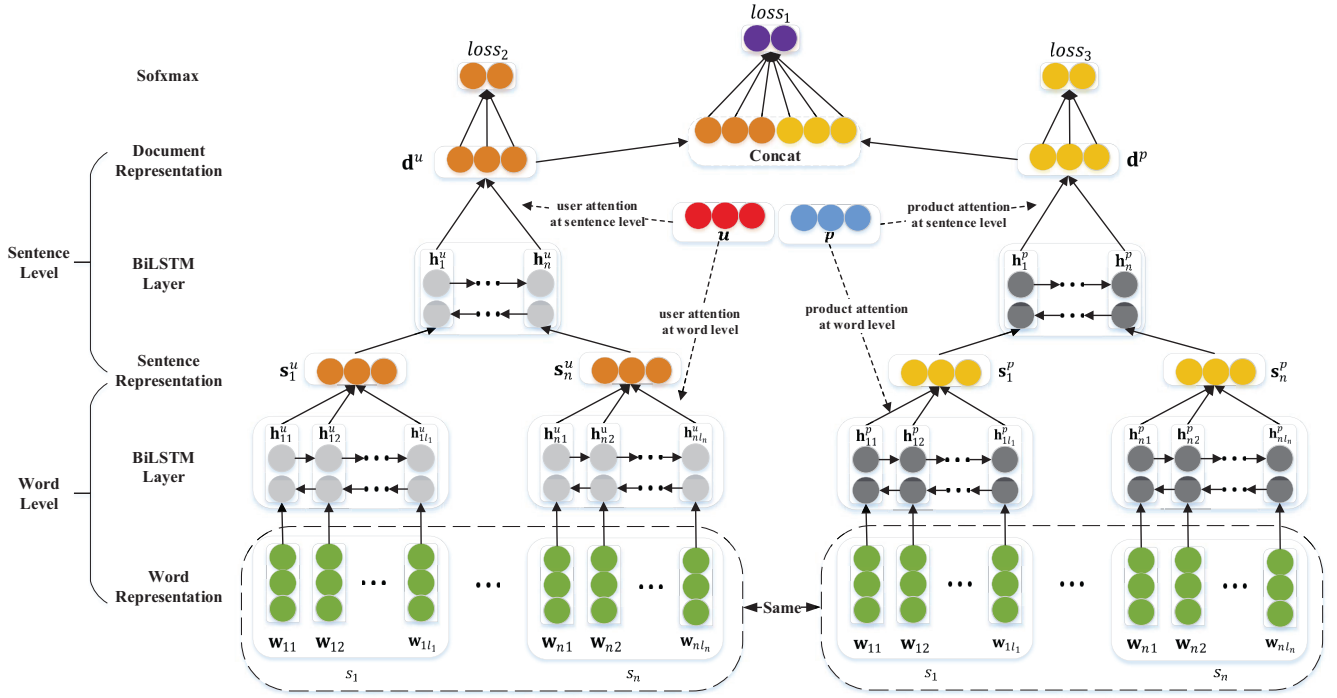
Figure 1: The architecture of Hierarchical User Attention and Product Attention neural network.

others tend to indicate product's characteristic. The two kinds of information play different roles in inferring the sentiment label of reviews, which implies a review has different latent semantic representations in user's view and product's view. In order to address the issue, we propose a novel framework to incorporate user and product information into sentiment classification.

Framework

We refer to our model as HUAPA for convenience. An illustration of HUAPA is shown in Figure 1. It consists of two components mainly: a hierarchical user attention network and a hierarchical product attention network. With attention mechanism, the former incorporates user information into review document modeling while the latter incorporates product information. Then, we concatenate the two review representations as the final representation to predict user's overall sentiment on a review about a product. In addition, we design a combined strategy to enhance review representations for sentiment classification. Specifically, we add a softmax classifier respectively to three review representation $\mathbf{d}^u$, $\mathbf{d}^p$ and $\mathbf{d}$, and introduce a weighted loss function as optimization objective, which proves effective. We will present the details of HUAPA in the following sections.

Hierarchical User Attention

From the user's point of view, not all words reflect equally user's preference or sentiment. In order to address the issue, we use user attention mechanism to extract user-specific

words that are significant to the meaning of sentence. Finally, the sentence representation is aggregated by the representations of those informative words. Formally, the enhanced sentence representation $\mathbf{s}_i^u$ is a weighted sum of word-level hidden states in user's view as:

$$\mathbf{s}_i^u = \sum_{j=1}^{l_i} \alpha_{ij}^u \mathbf{h}_{ij}^u, \tag{5}$$

where $\mathbf{h}_{ij}^u$ is the hidden state of the j-th word in the i-th sentence, α_{ij}^u is the attention weight of $\mathbf{h}_{ij}^u$ and measures the importance of the j-th word for current user. We map each user u into a continuous and real valued vector $\mathbf{u} \in \mathbb{R}^{d_u}$, where d_u denotes the dimension of user embeddings. Specifically, the attention weight α_{ij}^u for each hidden state can be defined as:

$$e(\mathbf{h}_{ij}^u, \mathbf{u}) = (\mathbf{v}_w^u)^\top \tanh\left(\mathbf{W}_{wh}^u \mathbf{h}_{ij}^u + \mathbf{W}_{wu}^u \mathbf{u} + \mathbf{b}_w^u\right), \tag{6}$$

$$\alpha_{ij}^u = \frac{\exp\left(e\left(\mathbf{h}_{ij}^u, \mathbf{u}\right)\right)}{\sum_{k=1}^{l_i} \exp\left(e\left(\mathbf{h}_{ik}^u, \mathbf{u}\right)\right)} \tag{7}$$

where $\mathbf{v}_w^u$ is weight vector and $(\mathbf{v}_w^u)^\top$ represents its transpose, $\mathbf{W}_{wh}^u$ and $\mathbf{W}_{wu}^u$ are weight matrices, $e(\cdot)$ is a score function which scores the importance of words for composing sentence representation about current user.

Similarly, different sentences contribute unequally to document semantics for users. Therefore, in sentence level, we also use a attention mechanism with user vector $\mathbf{u}$ in word level to generate the document representation. The docu-

Datasets	#classes	#docs	#users	#products	#docs/user	#docs/product	#sens/doc	#words/sen
IMDB	10	84,919	1,310	1,635	64.82	51.94	16.08	24.54
Yelp 2013	5	78,966	1,631	1,633	48.42	48.36	10.89	17.38
Yelp 2014	5	231,163	4,818	4,194	47.97	55.11	11.41	17.26

Table 1: Statistics of IMDB, Yelp2013 and Yelp2014 datasets

ment representation $\mathbf{d}^u$ in user's view is obtained via:

$$e\left(\mathbf{h}_i^u, \mathbf{u}\right) = \left(\mathbf{v}_s^u\right)^\top \tanh\left(\mathbf{W}_{sh}^u \mathbf{h}_i^u + \mathbf{W}_{su}^u \mathbf{u} + \mathbf{b}_s^u\right), \quad (8)$$

$$\beta_i^u = \frac{\exp\left(e\left(\mathbf{h}_i^u, \mathbf{u}\right)\right)}{\sum_{k=1}^n \exp\left(e\left(\mathbf{h}_k^u, \mathbf{u}\right)\right)}, \quad (9)$$

$$\mathbf{d}^u = \sum_{i=1}^n \beta_i^u \mathbf{h}_i^u, \quad (10)$$

where $\mathbf{h}_i^u$ is the hidden state of the i-th sentence in a review document, β_i^u is the weight of hidden state $\mathbf{h}_i^u$ in sentence level and can be calculated similar to the word level.

Hierarchical Product Attention

It is also true for different products that every word or sentence contributes different information to the text semantics. Based on the common sense, hierarchical product attention incorporates product information into review representation similar to hierarchical user attention. In product's view, the sentence representation $\mathbf{s}_i^p$ and document representation $\mathbf{d}^p$ of a review are obtained formally as:

$$\mathbf{s}_i^p = \sum_{j=1}^{l_i} \alpha_{ij}^p \mathbf{h}_{ij}^p, \quad (11)$$

$$\mathbf{d}^p = \sum_{i=1}^n \beta_i^p \mathbf{h}_i^p, \quad (12)$$

where α_{ij}^p and β_i^p are the weight of hidden state $\mathbf{h}_{ij}^p$ in word level and $\mathbf{h}_i^p$ in sentence level respectively.

Combined Strategy

In order to make full use of document representation $\mathbf{d}^u$ and $\mathbf{d}^p$, we design a combined strategy for training and the final prediction.

Since document representation $\mathbf{d}^u$ and $\mathbf{d}^p$ are high level representations of review in user's view and product's view respectively. Hence, we concatenate them as the final review representation for sentiment classification without feature engineering:

$$\mathbf{d} = [\mathbf{d}^u; \mathbf{d}^p]. \quad (13)$$

Specifically, we use a linear layer and a softmax layer to project review representation $\mathbf{d}$ into review sentiment distribution of C classes:

$$p = \text{softmax}\left(\mathbf{W}\mathbf{d} + \mathbf{b}\right). \quad (14)$$

In our model, the cross-entropy error between ground truth distribution of review sentiment and p is defined as $loss_1$:

$$loss_1 = -\sum_{d \in T} \sum_{c=1}^C p_c^g(d) \cdot \log\left(p_c(d)\right), \quad (15)$$

where p_c^g is the probability of sentiment label c with ground truth being 1 and others being 0, T represents the training set.

To make the review representations $\mathbf{d}^u$ and $\mathbf{d}^p$ both have certain predictive capability and futher improve performance for sentiment classification, we add a softmax classifier respectively to $\mathbf{d}^u$ and $\mathbf{d}^p$. The corresponding losses are defined as follows:

$$p^u = \text{softmax}\left(\mathbf{W}^u \mathbf{d}^u + \mathbf{b}^u\right), \quad (16)$$

$$loss_2 = -\sum_{d \in T} \sum_{c=1}^C p_c^g(d) \cdot \log\left(p_c^u(d)\right), \quad (17)$$

$$p^p = \text{softmax}\left(\mathbf{W}^p \mathbf{d}^p + \mathbf{b}^p\right), \quad (18)$$

$$loss_3 = -\sum_{d \in T} \sum_{c=1}^C p_c^g(d) \cdot \log\left(p_c^p(d)\right), \quad (19)$$

where p^u is the predicted sentiment distribution of with user's view and p^p is the predicted sentiment distribution of with product's view. The final loss of our model is a weighted sum of $loss_1$, $loss_2$ and $loss_3$ as:

$$L = \lambda_1 loss_1 + \lambda_2 loss_2 + \lambda_3 loss_3. \quad (20)$$

The $loss_2$ and $loss_3$ are introduced as supervised information to improve review representations and further help sentiment classification. Note that, we predict review sentiment label according to the distribution p because it contains both user information and product information.

Experiments

We conduct experiments on several real-world datasets to validate the effectiveness of our model and report empirical results in this section.

Experiments Settings

We conduct experiments on three sentiment classification datasets[2] with user and product information, which are from IMDB and Yelp Dataset Challenge in 2013 and 2014 (Tang, Qin, and Liu 2015b). The statistics of the datasets are summarized in Table 1. The datasets are split into three parts, 80% for training, 10% for validation, and the remaining 10% for test. We use standard $Accuracy$ to measure the overall sentiment classification performance, and $RMSE$ to measure the divergences between predicted sentiment label and

---

[2]http://ir.hit.edu.cn/ dytang/paper/acl2015/dataset.7z

Models	IMDB		Yelp 2013		Yelp 2014	
	Acc.	RMSE	Acc.	RMSE	Acc.	RMSE
Models without user and product information						
Majority	0.196	2.495	0.411	1.060	0.392	1.097
Trigram	0.399	1.783	0.569	0.814	0.577	0.804
TextFeature	0.402	1.793	0.556	0.845	0.572	0.800
AvgWordvec+SVM	0.304	1.985	0.526	0.898	0.530	0.893
SSWE+SVM	0.312	1.973	0.549	0.849	0.557	0.851
Paragraph Vector	0.341	1.814	0.554	0.832	0.564	0.802
RNTN+Recurrent	0.400	1.764	0.574	0.804	0.582	0.821
UPNN(CNN and no UP)	0.405	1.629	0.577	0.812	0.585	0.808
NSC	0.443	1.465	0.627	0.701	0.637	0.686
NSC+LA	0.487	1.381	0.631	0.706	0.630	0.715
NSC+LA(BiLSTM)	0.490	1.325	0.638	0.691	0.646	0.678
Models with user and product information						
Trigram+UPF	0.404	1.764	0.570	0.803	0.576	0.789
TextFeature+UPF	0.402	1.774	0.561	1.822	0.579	0.791
JMARS	N/A	1.773	N/A	0.985	N/A	0.999
UPNN(CNN)	0.435	1.602	0.596	0.784	0.608	0.764
UPNN(NSC)	0.471	1.443	0.631	0.702	N/A	N/A
LUPDR	0.488	1.451	0.639	0.694	0.639	0.688
NSC+UPA	0.533	1.281	0.650	0.692	0.667	0.654
NSC+UPA(BiLSTM)	0.529	1.247	0.655	0.672	0.669	0.654
HUAPA	**0.550**	**1.185**	**0.683**	**0.628**	**0.686**	**0.626**

Table 2: Reviews sentiment classification results. Acc.(Accuracy, higher is better) and RMSE(lower is better) are the evaluation metrics. The best performances are in bold. HUAPA outperforms previous best state-of-the-art method significantly($p < 0.01$).

ground truth label. They are defined as follows:

$$Accuracy = \frac{T}{N}, \tag{21}$$

$$RMSE = \sqrt{\frac{\sum_{k=1}^{N}(gd_k - pr_k)^2}{N}}, \tag{22}$$

where T is the numbers of predicted sentiment labels that are same with ground truth sentiment labels of reviews, N is the numbers of review documents, gd_k represents the ground sentiment label, and pr_k denotes predicted sentiment label.

We pre-train the 200-dimensional word embeddings on each dataset with SkipGram (Mikolov et al. 2013). The word embeddings are not fine-tuned when training, so hierarchical user attention and hierarchical product attention use same word embeddings. We set the user embeddings dimension and product embeddings dimension to be 200, and randomly initialize them from a uniform distribution $U(-0.01, 0.01)$. The dimensions of hidden states in LSTM cell are set to 100. In this setting, a bidirectional LSTM gives us 200 dimensional output for word/sentence representation. To speed up training, we limit that a review document has 40 sentences at most and every sentence has no more than 50 words. We use Adam (Kingma and Ba 2015) to update parameters when training and empirically set initial learning rate to be 0.005. Finally, We select the best parameters based on performance on the validation set, and evaluate the parameters on the test set. Note that, we do not use any regularization or dropout (Srivastava et al. 2014) to improve performance of the model.

Baselines

We compare our model HUAPA with several baseline methods for document-level review sentiment classification:

Majority assigns the majority sentiment label in training set to each review document in test set.

Trigram uses unigrams, bigrams and trigrams as features to train a SVM classifier with LibLinear (Fan et al. 2008).

TextFeature extracts sophisticated text features to train SVM classifier, such word/character n-grams, sentiment lexicon features, cluster features, etc. (Kiritchenko, Zhu, and Mohammad 2014).

UPF extracts user leniency and corresponding product popularity features (Gao et al. 2013) from training data, and further concatenates them with the features in **Trigram** and **TextFeature**.

AvgWordvec averages word embeddings of a document to generate document representation, then feeds it into a SVM classifier as features.

SSWE learns sentiment-specific word embeddings (SSWE) (Tang et al. 2014), and uses max/min/average pooling to obtain document representation which is used as features for a SVM classifier.

RNTN + RNN uses the Recursive Neural Tensor Network (RNTN) (Socher et al. 2013) to obtain sentence representations, then feeds them into the Recurrent Neural Network (RNN). Afterwards, the hidden vectors of RNN are averaged to generate document representation for sentiment classification.

Paragraph Vector implements the Distributed Memory Model of Paragraph Vectors (Le and Mikolov 2014) for doc-

Models	IMDB		Yelp 2013		Yelp 2014	
	Acc.	RMSE	Acc.	RMSE	Acc.	RMSE
NSC+LA(BiLSTM)	0.490	1.325	0.638	0.691	0.646	0.678
HUA	0.521	1.300	0.649	0.691	0.663	0.661
HPA	0.493	1.326	0.641	0.681	0.646	0.678
HUAPA	**0.550**	**1.185**	**0.683**	**0.628**	**0.686**	**0.626**

Table 3: Effect of user attention and product attention. HUA only uses user information and local text, and HPA only uses product information and local text.

λ_1	λ_2	λ_3	IMDB		Yelp 2013		Yelp 2014	
			Acc.	RMSE	Acc.	RMSE	Acc.	RMSE
1.0	0.0	0.0	0.538	1.229	0.669	0.658	0.675	0.647
0.7	0.3	0.0	0.541	1.239	0.672	0.644	0.680	0.641
0.7	0.0	0.3	0.540	1.287	0.675	0.646	0.679	0.633
0.4	0.3	0.3	**0.550**	**1.185**	**0.683**	**0.628**	**0.686**	**0.626**

Table 4: Effect of the different weighted loss.

ument sentiment classification. The window size is tuned on validation set.

JMARS is a recommendation algorithm (Diao et al. 2014), which uses the information of users and aspects with collaborative filtering and topic modeling to predict document sentiment rating.

UPNN introduces a word-level preference matrix and a representation vector for each user and each product into CNN sentiment classifier (Kim 2014). The meaning of words can modified in the input layer with the preference matrix. Finally, it concatenates the user/product representation vectors with generated review representation as features fed into softmax layer (Tang, Qin, and Liu 2015b).

LUPDR uses recurrent neural network to embed temporal relations of reviews into the categories of distributed user and product representations learning for the sentiment classification of reviews (Chen et al. 2016b).

NSC uses hierarchical LSTM model to encode review text for sentiment classification.

NSC+LA implements the idea of local semantic attention (Yang et al. 2016) based on **NSC**.

NSC+UPA puts user and product information account together and uses hierarchical LSTM model with attention mechanism to generate a review representation for sentiment classification.

Chen et al. (2016a) do not implement the model **NSC+UPA** and **NSC+LA** with bidirectional LSTM. To make the experimental results more convincing, we implement and train them in our experimental settings. In addition to **LUPDR** and the models related to **NSC**, we report the results in (Tang, Qin, and Liu 2015b) since we use the same datasets for other baseline methods above.

Model Comparisons

The experimental results are given in Table 2, which are divided into two parts: the models only using the local text information, and the models incorporating both local text information and the global user and product information.

From the first part, we can see that the majority performs very poor because it does not use any text, user, and product information. Compared to the methods taking SVM as classifier, hierarchical neural networks achieve better performances generally. In addition, the results show NSC+LA obtains a considerable improvements based on NSC, which proves that importance of selecting more meaningful words and sentences in sentiment classification. It is also a main reason that attention mechanism is introduced into sentiment classification.

From the second part, we observe that the methods considering user and product information achieve more or less improvements compared to the corresponding methods in the first part. For example, TextFeature+UPF achieves 0.5% improvement and NSC+UPA obtains 2.3% improvement on Yelp2013 in accuracy. The comparisons indicate that the user and product information is helpful for sentiment classification.

The experimental results show that our proposed model with user attention and product attention achieves best performance on all datasets. We can see improvements regardless of data scale. For smaller dataset such as Yelp2013 and IMDB, our model outperforms the previous best state-of-the-art method by 2.8% and 1.7% respectively in accuracy. This finding is consistent on larger dataset. Our model achieves improvement by 1.7% on dataset Yelp2014 in accuracy. The observations demonstrate that our model incorporates user and product information in a more effective way, which finally improves review representations for sentiment classification.

Model Analysis: Effect of User Attention and Product Attention

To investigate the effects of single user attention or product attention, we also implement independent hierarchical user attention network (HUA) and hierarchical product attention network (HPA). Table 3 shows the performance of single attention mechanism with user or product information. From the table, we can observe that:

- Compared to the model NSC+LA(BiLSTM) only using local semantic attention, HUA and HPA both achieve some improvements, which validates the rationality of incorporating user and product into sentiment classification via attention mechanism. The results also indicate that user attention or product attention can capture more information related to sentiment.

- The user information is more effective than the product information to enhance review representations. Although some words or sentences in reviews show product's characteristic, the ratings are finally decided by users. Hence, it is reasonable that the discrimination of user's preference is more obvious than product's characteristic.

- Compared to single user attention or product attention, our model achieves better performance, which indicates that user and product information both contribute to our model. The results demonstrate our user attention and product attention mechanism can catch the specific user's preference and product's characteristic.

Model Analysis: Effect of the Different Weighted Loss

The λ_1, λ_2, and λ_3 respectively represent the weight of $loss_1$, $loss_2$ and $loss_3$. We investigate the effect of different weighted loss by empirically adjusting their proportion. When λ_2 is set to 0, we do not use $loss_2$ to enhance the review representations. Similarly, we set λ_3 to 0 to avoid the effect of $loss_3$. The experimental results are in Table 4. From the Table 4 and Table 2, we can observe that:

- Compared to other state-of-the-art methods, our model without $loss_2$ and $loss_3$ also achieve consistent improvements on the three datasets. It indicates that our attention mechanism is more effective in incorporating user information and product information.

- When considering $loss_2$ or $loss_3$, HUAPA both obtains some improvements. It is obvious that full HUAPA model achieves best performance. The results demonstrate that with the designed combined strategy, better review representations from two views can be achieved and further help sentiment classification.

Case Study for Visualization of Attention

To validate our observation and demonstrate the effectiveness of user attention and product attention, we take some review instances in Yelp2013 dataset as example. We visualize the attention weights of these reviews at word level in HUAPA. The results are shown in Figure 2. Note that, the darker color means higher weight.

For the review 1, we can see that the word "*love*" has highest attention weight in user attention and the word "*cool*" has highest attention weight in product attention. In fact, it is intuitive that "*love*" usually expresses user's affection or preference, and "*cool*" is used to describe the characteristic of product. There are also some reviews that user's preference is inconsistent with product's characteristic, such as review 3 in Figure 2. The content of the review 3 is "*much of it was quite good but I was disappointed with the spider roll.*". It is
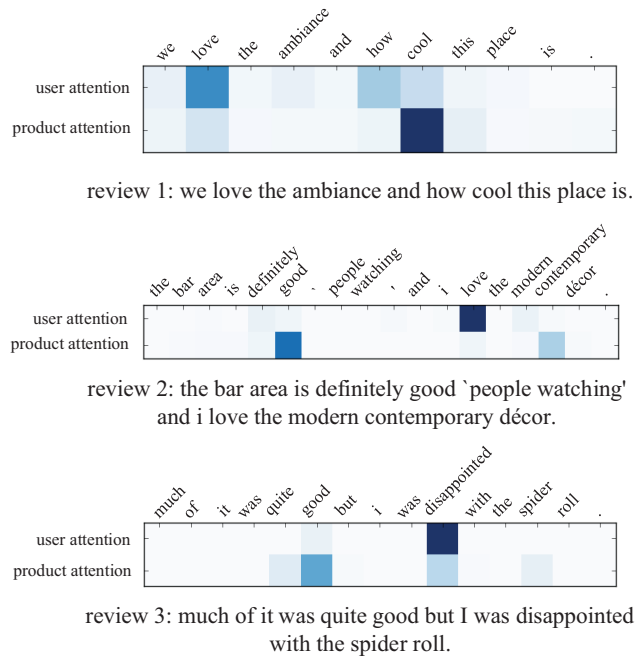

review 1: we love the ambiance and how cool this place is.

review 2: the bar area is definitely good `people watching' and i love the modern contemporary décor.

review 3: much of it was quite good but I was disappointed with the spider roll.

Figure 2: Visualization of user attention and product attention over words

obvious that the word "*good*" indicates the product's characteristic and the word "*disappointed*" shows user's negative sentiment. Our model not only catches such information but also gives right prediction. The visualizations of attention demonstrate that our model can capture global user's preference and product's characteristic.

Conclusion

In this paper, we propose a novel framework incorporating user and product information for sentiment classification. Specifically, we firstly apply two individual hierarchical neural networks to generate two representations, with user attention or with product attention. Then, we design a combined strategy to make full use of the two representations for training and final prediction. We evaluate our model on several sentiment classification datasets. The experimental results show that our model achieves obvious and consistent improvements compared to other state-of-the-art methods. Finally, the visualizations of attention also show our model can capture user information and product information.

Acknowledgments

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Bengio, Y.; Ducharme, R.; Vincent, P.; and Jauvin, C. 2003. A neural probabilistic language model. *JMLR* 3(Feb):1137–1155.

Chen, H.; Sun, M.; Tu, C.; Lin, Y.; and Liu, Z. 2016a. Neural sentiment classification with user and product attention. In *EMNLP*, 1650–1659.

Chen, T.; Xu, R.; He, Y.; Xia, Y.; and Wang, X. 2016b. Learning user and product distributed representations using a sequence model for sentiment analysis. *IEEE Computational Intelligence Magazine* 11(3):34–44.

Dahl, G. E.; Yu, D.; Deng, L.; and Acero, A. 2012. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *TASLP* 20(1):30–42.

Diao, Q.; Qiu, M.; Wu, C.-Y.; Smola, A. J.; Jiang, J.; and Wang, C. 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In *KDD*, 193–202.

Ding, X.; Liu, B.; and Yu, P. S. 2008. A holistic lexicon-based approach to opinion mining. In *WSDM*, 231–240.

Fan, R.-E.; Chang, K.-W.; Hsieh, C.-J.; Wang, X.-R.; and Lin, C.-J. 2008. Liblinear: A library for large linear classification. *JMLR* 9(Aug):1871–1874.

Gao, W.; Yoshinaga, N.; Kaji, N.; and Kitsuregawa, M. 2013. Modeling user leniency and product popularity for sentiment classification. In *IJCNLP*, 1107–1111.

Glorot, X.; Bordes, A.; and Bengio, Y. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *ICML*, 513–520.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation* 9(8):1735–1780.

Johnson, R., and Zhang, T. 2015. Effective use of word order for text categorization with convolutional neural networks. In *HLT-NAACL*, 103–112.

Kim, Y. 2014. Convolutional neural networks for sentence classification. In *EMNLP*, 1746–1751.

Kingma, D., and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Kiritchenko, S.; Zhu, X.; and Mohammad, S. M. 2014. Sentiment analysis of short informal texts. *JAIR* 50:723–762.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*, 1097–1105.

Le, Q., and Mikolov, T. 2014. Distributed representations of sentences and documents. In *ICML*, 1188–1196.

Liu, B. 2012. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies* 5(1):1–167.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*, 3111–3119.

Pang, B., and Lee, L. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL*, 115–124.

Pang, B., and Lee, L. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval* 2(12):1–135.

Pang, B.; Lee, L.; and Vaithyanathan, S. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *EMNLP*, 79–86.

Qu, L.; Ifrim, G.; and Weikum, G. 2010. The bag-of-opinions method for review rating prediction from sparse text patterns. In *COLING*, 913–921.

Socher, R.; Pennington, J.; Huang, E. H.; Ng, A. Y.; and Manning, C. D. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *EMNLP*, 151–161.

Socher, R.; Huval, B.; Manning, C. D.; and Ng, A. Y. 2012. Semantic compositionality through recursive matrix-vector spaces. In *EMNLP-CoNLL*, 1201–1211.

Socher, R.; Perelygin, A.; Wu, J. Y.; Chuang, J.; Manning, C. D.; Ng, A. Y.; Potts, C.; et al. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, 1631–1642.

Srivastava, N.; Hinton, G. E.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *JMLR* 15(1):1929–1958.

Sukhbaatar, S.; Weston, J.; Fergus, R.; et al. 2015. End-to-end memory networks. In *NIPS*, 2440–2448.

Taboada, M.; Brooke, J.; Tofiloski, M.; Voll, K.; and Stede, M. 2011. Lexicon-based methods for sentiment analysis. *Computational Linguistics* 37(2):267–307.

Tang, D.; Wei, F.; Yang, N.; Zhou, M.; Liu, T.; and Qin, B. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *ACL*, 1555–1565.

Tang, D.; Qin, B.; and Liu, T. 2015a. Document modeling with gated recurrent neural network for sentiment classification. In *EMNLP*, 1422–1432.

Tang, D.; Qin, B.; and Liu, T. 2015b. Learning semantic representations of users and products for document level sentiment classification. In *ACL*, 1014–1023.

Vinyals, O.; Kaiser, Ł.; Koo, T.; Petrov, S.; Sutskever, I.; and Hinton, G. 2015. Grammar as a foreign language. In *NIPS*, 2773–2781.

Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2048–2057.

Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A. J.; and Hovy, E. H. 2016. Hierarchical attention networks for document classification. In *HLT-NAACL*, 1480–1489.